MECHANICS BASED DESIGN OF STRUCTURES AND MACHINES
2026, VOL. 54, NO. 1, 2623082
https://doi.org/10.1080/15397734.2026.2623082

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS Check for updates

Non-iterative implicit integration methods with linearization-based equations of motion approximation for real-time multibody vehicle dynamics simulation

Jan-Lukas Archut and Burkhard Corves

Institute of Mechanism Theory, Machine Dynamics and Robotics, RWTH Aachen University, Aachen, Germany

ABSTRACT

The growing importance of virtual vehicle development and subjective assessments of driving characteristics on driving simulators pushes the demand for real-time vehicle dynamics simulation. Real-time capable high-fidelity multibody simulation models are required that capture the influence of the kinematics and elastokinematics of wheel suspensions on driving dynamics accurately. Elastic bushings, commonly used in wheel suspensions to enhance ride comfort and handling, introduce high numerical stiffness of the equations of motion and pose significant challenges for real-time simulation. Non-iterative implicit integration methods offer an attractive solution by combining stability, accuracy and deterministic computational effort. These methods avoid iterative solutions of nonlinear equation systems by exploiting the Jacobians of the equations of motion, making them well suited for real-time applications. To reduce their computational effort, this study simplifies several non-iterative implicit integration methods with a linearization-based equations of motion approximation that is updated every few integration steps. The performance of the methods is evaluated and compared for real-time multibody simulations of a double wishbone suspension with elastic bushings. The influence of the integration step size and the update frequency of the equations of motion evaluation and linearization on computation times and accuracy is analyzed using dynamic suspension tests.

ARTICLE HISTORY
Received 24 May 2025
Accepted 16 January 2026

KEYWORDS
multibody dynamics; linear-implicit integration; linear-implicit euler method; Rosenbrock-Wanner method; HHT-α integrator

1. Introduction

Virtual vehicle development is becoming increasingly important as it offers the potential for a more targeted, cost-effective and safer development process. The driving simulator is one of the key tools for driving dynamics evaluation, allowing for subjective assessments even in early development stages without the need for physical prototype vehicles (Carpinelli et al. 2016; Jafarzadehpour, Bidlingmaier, and Corves 2021; Schmidt et al. 2015). This requires real-time capable vehicle dynamics simulation models that provide their output at a fixed output rate and complete all computations in each simulation step within a predefined time frame. Furthermore, high accuracy is required to realistically capture the driving characteristics of the vehicle to be assessed. In the following, we will focus on the wheel suspension system, as it significantly affects

CONTACT Jan-Lukas Archut archut@igmr.rwth-aachen.de Institute of Mechanism Theory, Machine Dynamics and Robotics, RWTH Aachen University, Eilfschornsteinstrasse 18, Aachen 52062, Germany.

Supplemental data for this article can be accessed online at https://doi.org/10.1080/15397734.2026.2623082.
Communicated by Davide Tavernini.

© 2026 The Author(s). Published with license by Taylor & Francis Group, LLC
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.

driving dynamics through its kinematic and elastokinematic design. Elastic bushings are used to connect the suspension bodies in order to improve ride comfort and driving characteristics (Ersoy and Gies 2017). They lead to compliant wheel position changes that affect the handling and are therefore important to include in accurate vehicle dynamics models.

There are many approaches in literature for real-time vehicle dynamics simulation. A systematic overview is provided in (Archut and Corves, 2025). Lumped-mass and lookup-table based models simplify the suspension to a single mass that performs only translational wheel travel motions with respect to the vehicle body (Blundell and Harty 2015). Suspension kinematics and elastokinematics are optionally represented by means of static lookup-tables (Adamski 2014; Candreva et al. 2014). This modeling approach leads to simple, inherently real-time capable vehicle dynamics models, but their validity for detailed analysis is limited (Cibrario et al. 2014; Falchi et al. 2022). In contrast, multibody-based suspension models describe the suspension as a complex multibody system composed of the individual suspension components, allowing for a detailed representation of its kinematics and elastokinematics (Blundell and Harty 2015; Rill 2019). However, incorporating elastic bushings in vehicle dynamics simulation models for real-time applications is challenging. Their high stiffness and damping properties, combined with the low inertia of the suspension control arms and other components, lead to highly oscillatory or strongly damped motions. These "fast" motions superpose motions on the "slow" time scale, e.g., the comparatively slow wheel travel motion. This is associated with high numerical stiffness of the equations of motion (EOM), posing special challenges for real-time capable numerical integration. Explicit integration methods are often not suited due to their limited stability range. While implicit integration methods offer better stability properties, they usually require solving of nonlinear equations through iterative solution methods, typically involving high and non-deterministic computational effort. This limits their applicability for real-time simulations.

Several approaches in literature aim to enable real-time simulations of multibody-based suspension models with elastic bushings (Archut and Corves, 2025). Many of these methods rely on model simplifications to reduce complexity and numerical stiffness, e.g. by superimposing elastokinematic effects based on quasistatic approximations (Salaani, Heydinger, and Grygier 2003; Rulka and Pankiewicz 2005; Kim and Jeong 2009) or data-driven models using neural networks (Zhang et al. 2025), neglecting the inertia of fast-moving small masses (Burgermeister, Arnold, and Eichberger 2011), incorporating additional spring-damper elements (Lefèvre, Kracht, and Schramm 2020) or applying low-pass filters (Lefèvre, Kracht, and Schramm 2020; Kracht and Schramm 2020). These modifications often enable the use of explicit integration methods but may come at the cost of reduced model accuracy. Other approaches focus on improving computational efficiency without compromising accuracy as much. This includes approximating constraint equations (Lee, Kang, and Bae 2014), utilizing efficient linear algebra implementations and matrix libraries (Hidalgo and García de Jalón 2015) as well as parallel processing (Schmidt et al. 2015; Hidalgo and García de Jalón 2015; Prescott 2014; Klomp, Sundström, and Johnsson 2017), or limiting the number of iterations when solving nonlinear equation systems to make the computational effort of implicit integration methods deterministic (Prescott 2014).

Another approach are non-iterative implicit integration methods such as linear-implicit integration schemes, which promise stable and real-time capable time integration of numerically stiff systems (Archut and Corves, 2025). They utilize the Jacobians of the EOM to transform the nonlinear equation systems of implicit integration methods into linear equation systems (Rill 2006), which can be solved efficiently and with deterministic computational effort using direct methods. Hence, these methods are well suited for real-time applications. Numerical stability can also be achieved with approximated Jacobians instead of exact ones (Arnold, Burgermeister, and Eichberger 2007), which, e.g., are precomputed for characteristic configurations before real-time simulation (Arnold, Burgermeister, and Eichberger 2007) or are calculated neglecting the partial derivatives of the inertia forces (Rill 2006; Schmitt 2019). The linear-implicit Euler method is a non-iterative implicit

integration method frequently used in real-time vehicle dynamics simulation (Carpinelli et al. 2016; Rill 2006; Schmitt 2019; Butz, Simeon, and Stadler 2010). It is easy to implement but may lack sufficient accuracy for complex applications (Arnold, Burgermeister, and Eichberger 2007). An alternative are higher-order Rosenbrock-Wanner (ROW) methods or W-methods (Rosenbrock 1963; Steihaug and Wolfbrandt 1979), which can be considered as a linear-implicit version of Runge-Kutta methods. Their computational effort, order and numerical damping can be adapted through specifying their number of stages and parameters (Arnold, Burgermeister, and Eichberger 2007; Bursi et al. 2008; Rang 2015). A two-stage ROW integration approach has been utilized for real-time simulation of detailed suspension models with different topologies including elastic bushings (Archut and Corves, 2025). More recently, a novel non-iterative version of the structural implicit HHT-α integrator has been proposed for real-time simulations, whose numerical damping properties can easily be adjusted through the numerical damping parameter α (Kim, Song, and Kim 2019). Its applicability has been demonstrated for a simple multibody system (Kim, Song, and Kim 2019).

Against this background, the goal of this paper is twofold: it aims for a modification and a more detailed analysis of the aforementioned non-iterative integration methods. To reduce the computational effort of these methods, the first goal is to simplify them by exploiting the linearization of the EOM. Usually, the EOM are evaluated once or even multiple times in each integration step. In the following, however, the EOM will be approximated based on a linearization that is computed analytically and updated only every few integration steps. In the intermediate integration steps, both the EOM and their Jacobians are approximated accordingly. This simplification seeks to scale computational effort and accuracy of the simulation by adjusting the frequency at which the EOM evaluation and linearization is updated. It will be applied to different non-iterative integration methods, including the commonly used linear-implicit Euler method, ROW/W-methods and the non-iterative implicit HHT-α method. The second goal of this paper is to compare these simplified methods with each other. Their performance in real-time vehicle dynamics simulations will be evaluated using a multibody simulation model of a representative double wishbone suspension with elastic bushings. Accuracy and computational effort of the different integration methods are compared and the influence of the integration step size as well as the EOM evaluation and linearization update frequency is analyzed.

The remainder of this paper is structured as follows: The considered non-iterative implicit integration methods are introduced and simplified using the linearization-based EOM approximation in Sect. 2. The multibody simulation model of the wheel suspension is presented in Sect. 3. In Sect. 4, the dynamic suspension tests used to analyze the performance of the integration methods are described. The results in terms of accuracy and computation times are presented in Sect. 5. The paper ends with the conclusion given in Sect. 6.

2. Non-iterative implicit integration methods for multibody simulation

The EOM of a multibody system described by a minimum set of coordinates are given by

$$\dot{\mathbf{r}} = \mathbf{K}(\mathbf{r})\mathbf{v} \tag{1a}$$

$$\mathbf{M}(\mathbf{r})\dot{\mathbf{v}} = \mathbf{h}(\mathbf{r}, \mathbf{v}, \mathbf{u}) \tag{1b}$$

with the initial conditions $\mathbf{r}(t = 0) = \mathbf{r}_0$ and $\mathbf{v}(t = 0) = \mathbf{v}_0$. They are a set of nonlinear ordinary differential equations for the vector of generalized coordinates $\mathbf{r}$ and the vector of generalized velocities $\mathbf{v}$. $\mathbf{M}(\mathbf{r})$ denotes the mass matrix, which may depend on the generalized coordinates, $\mathbf{h}$ is the vector of generalized forces, and $\mathbf{K}(\mathbf{r})$ is the kinematics matrix describing the linear transformation of the generalized velocities to the time derivatives of the generalized coordinates. The variable $\mathbf{u}$ serves as a placeholder for all possible inputs, e.g., forces or prescribed motions. Further explicit time-dependencies are not considered.

The EOM (1) can be linearized in an arbitrary linearization state $\mathbf{r}_{\text{lin}}$, $\mathbf{v}_{\text{lin}}$ and $\mathbf{u}_{\text{lin}}$ and approximated as

$$\dot{\mathbf{r}} = \mathbf{K}_{\text{lin}}\mathbf{v} \tag{2a}$$

$$\mathbf{M}_{\text{lin}}\dot{\mathbf{v}} = \mathbf{h}_{\text{lin}} + \mathbf{J}_{\text{r,lin}}(\mathbf{r} - \mathbf{r}_{\text{lin}}) + \mathbf{J}_{\text{v,lin}}(\mathbf{v} - \mathbf{v}_{\text{lin}}) + \mathbf{J}_{\text{u,lin}}(\mathbf{u} - \mathbf{u}_{\text{lin}}) \tag{2b}$$

with the EOM evaluation in the linearization state according to

$$\mathbf{h}_{\text{lin}} = \mathbf{h}(\mathbf{r}_{\text{lin}}, \mathbf{v}_{\text{lin}}, \mathbf{u}_{\text{lin}}) \tag{3a}$$

$$\mathbf{M}_{\text{lin}} = \mathbf{M}(\mathbf{r}_{\text{lin}}) \tag{3b}$$

$$\mathbf{K}_{\text{lin}} = \mathbf{K}(\mathbf{r}_{\text{lin}}) \tag{3c}$$

and the Jacobians

$$\mathbf{J}_{\text{r,lin}} = \frac{\partial \mathbf{h}}{\partial \mathbf{r}} \tag{4a}$$

$$\mathbf{J}_{\text{v,lin}} = \frac{\partial \mathbf{h}}{\partial \mathbf{v}} \tag{4b}$$

$$\mathbf{J}_{\text{u,lin}} = \frac{\partial \mathbf{h}}{\partial \mathbf{u}} \quad . \tag{4c}$$

The numerical stiffness of the multibody system is quantified based on the eigenvalues $\lambda_i \in \mathbb{C}$, $i = 1, \ldots, n$, of the linearized EOM, using the stiffness ratio in the linearization state defined as (Dahmen and Reusken 2008)

$$\text{SR} = \max_{i,j}(|\lambda_i|/|\lambda_j|) \quad . \tag{5}$$

The system is called numerically stiff if the real parts of all eigenvalues λ_i are negative and the stiffness ratio is much larger than one (Dahmen and Reusken 2008). This implies that the system exhibits very different stiffness or damping properties, leading to dynamics on different time scales, i.e., "slow" and "fast" dynamics. Accordingly, multibody systems of wheel suspensions with elastic bushings have high numerical stiffness, cf. Sect. 1.

In the following, different non-iterative implicit time integration methods that promise stable and real-time capable integration of the numerically stiff EOM (1) will be introduced and simplified by exploiting the linearization (2). Table 1 shows an overview of the considered integration methods, namely the linear-implicit Euler method, ROW-/W-methods and the non-iterative implicit HHT-α method with different parameters.

2.1. Linear-implicit Euler method

The well-known implicit Euler method can be written for EOM (1) as

$$\mathbf{r}^{k+1} = \mathbf{r}^k + \Delta t_{\text{int}}\mathbf{K}(\mathbf{r}^{k+1})\mathbf{v}^{k+1} \tag{6a}$$

$$\mathbf{M}(\mathbf{r}^{k+1})\mathbf{v}^{k+1} = \mathbf{M}(\mathbf{r}^{k+1})\mathbf{v}^k + \Delta t_{\text{int}}\mathbf{h}(\mathbf{r}^{k+1}, \mathbf{v}^{k+1}, \mathbf{u}^{k+1}), \tag{6b}$$

Table 1. Non-iterative implicit integration methods und parameter sets.

Nr.	Abbreviation	Method	Parameters
1)	LIE	Linear-implicit Euler method	–
2)	LSRT2	ROW-/W-method with two stages (W2)	(Bursi et al. 2008)
3)	ABE2	ROW-/W-method with two stages (W2)	(Arnold et al. 2007)
2)	LSRT3	ROW-/W-method with three stages (W3)	(Bursi et al. 2008)
5)	ROS3PRw	ROW-/W-method with three stages (W3)	(Rang 2015)
6)	nHHT-α (0)	Non-iterative implicit HHT-α method (nHHT-α)	$\alpha = 0$
7)	nHHT-α (−1/3)	Non-iterative implicit HHT-α method (nHHT-α)	$\alpha = -1/3$
8)	nHHT-α (−0.1)	Non-iterative implicit HHT-α method (nHHT-α)	$\alpha = -0.1$

with Δt_{int} denoting the fixed integration step size and k the current integration step. Based on that, the linear-implicit version of this method can be derived by exploiting the Jacobians (4) of the EOM (Rill 2006). To this end, we approximate the EOM in the next integration step $k+1$ using their linearization (3). This allows to transform the implicit nonlinear equation system (6b) into the linear equation system

$$\begin{aligned}\left(\mathbf{M}_{\text{lin}} - \Delta t_{\text{int}}\mathbf{J}_{\text{v,lin}} - \Delta t_{\text{int}}^2\mathbf{J}_{\text{r,lin}}\mathbf{K}_{\text{lin}}\right)\mathbf{v}^{k+1} &= \mathbf{M}_{\text{lin}}\mathbf{v}^k \\ +\Delta t_{\text{int}}\Big(\mathbf{h}_{\text{lin}} + \mathbf{J}_{\text{r,lin}}(\mathbf{r}^k - \mathbf{r}_{\text{lin}}) - \mathbf{J}_{\text{v,lin}}\mathbf{v}_{\text{lin}} + \mathbf{J}_{\text{u,lin}}(\mathbf{u}^{k+1} - \mathbf{u}_{\text{lin}})\Big).\end{aligned} \tag{7}$$

As a result, the generalized velocities $\mathbf{v}^{k+1}$ are calculated by directly solving the linear equation system (7), without the necessity for iterative solution methods. Subsequently, the generalized coordinates $\mathbf{r}^{k+1}$ are explicitly calculated using Equation (6a) and the approximation $\mathbf{K}(\mathbf{r}^{k+1}) = \mathbf{K}_{\text{lin}}$. By reusing the EOM linearization (2) from the linearization state, in particular using the generalized forces $\mathbf{h}_{\text{lin}}$ and their Jacobians in equation system (7), the expensive EOM evaluation and linearization before each integration step can be avoided.

It should be noted that the inputs $\mathbf{u}^{k+1}$ in the integration step to be calculated must be known for setting up equation system (7). However, they are usually directly available or can be approximated accordingly, e.g., based on prescribed load cases or driving maneuvers in offline-simulations or based on the driver's input in a driving simulator.

2.2. ROW- and W-methods

ROW- and W-methods are a linear-implicit version of Runge-Kutta methods (Rosenbrock 1963). As in the linear-implicit Euler method, a linear equation system is obtained by exploiting the Jacobians of the EOM, whereby W-methods, unlike ROW-methods, use approximated Jacobians (Steihaug and Wolfbrandt 1979). The integration formula of ROW-/W-methods with s stages, based on (Arnold, Burgermeister, and Eichberger 2007; Rang 2015) and modified for EOM (1) with explicit consideration of the inputs $\mathbf{u}$, is

$$\begin{aligned}\left(\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{M}_{\text{lin}}\end{bmatrix} - \Delta t_{\text{int}}\gamma_{ii}\begin{bmatrix}\mathbf{0} & \mathbf{K}_{\text{lin}}\\ \mathbf{J}_{\text{r,lin}} & \mathbf{J}_{\text{v,lin}}\end{bmatrix}\right)\begin{bmatrix}\Delta\mathbf{r}^{k,i}\\ \Delta\mathbf{v}^{k,i}\end{bmatrix} &= \begin{bmatrix}\mathbf{K}_{\text{lin}}\tilde{\mathbf{v}}^{k,i}\\ \mathbf{h}(\tilde{\mathbf{r}}^{k,i}, \tilde{\mathbf{v}}^{k,i}, \tilde{\mathbf{u}}^{k,i})\end{bmatrix} \\ +\Delta t_{\text{int}}\begin{bmatrix}\mathbf{0} & \mathbf{K}_{\text{lin}}\\ \mathbf{J}_{\text{r,lin}} & \mathbf{J}_{\text{v,lin}}\end{bmatrix}\sum_{j=1}^{i-1}\left(\gamma_{ij}\begin{bmatrix}\Delta\mathbf{r}^{k,j}\\ \Delta\mathbf{v}^{k,j}\end{bmatrix}\right) &+ \Delta t_{\text{int}}\begin{bmatrix}\mathbf{0}\\ \mathbf{J}_{\text{u,lin}}\end{bmatrix}\sum_{j=1}^{i}\left(\gamma_{ij}\Delta\mathbf{u}^k\right)\end{aligned} \tag{8}$$

with $i = 1, \ldots, s$ and the identity matrix $\mathbf{I}$. The vector $\Delta\mathbf{u}^k$ is given by

$$\Delta\mathbf{u}^k = \frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{\Delta t_{\text{int}}} \quad , \tag{9}$$

and the stage vectors $\tilde{\mathbf{r}}^{k,i}$, $\tilde{\mathbf{v}}^{k,i}$ and $\tilde{\mathbf{u}}^{k,i}$ are defined as

$$\tilde{\mathbf{r}}^{k,i} = \mathbf{r}^k + \Delta t_{\text{int}} \sum_{j=1}^{i-1} (a_{ij} \Delta \mathbf{r}^{k,j}) \tag{10a}$$

$$\tilde{\mathbf{v}}^{k,i} = \mathbf{v}^k + \Delta t_{\text{int}} \sum_{j=1}^{i-1} (a_{ij} \Delta \mathbf{v}^{k,j}) \tag{10b}$$

$$\tilde{\mathbf{u}}^{k,i} = \mathbf{u}^k + c_i \Delta t_{\text{int}} \Delta \mathbf{u}^k. \tag{10c}$$

With the vectors $\Delta \mathbf{r}^{k,i}$ and $\Delta \mathbf{v}^{k,i}$ from each stage i, the generalized coordinates $\mathbf{r}^{k+1}$ and velocities $\mathbf{v}^{k+1}$ are calculated as

$$\mathbf{r}^{k+1} = \mathbf{r}^k + \Delta t_{\text{int}} \sum_{i=1}^{s} (b_i \Delta \mathbf{r}^{k,i}) \tag{11a}$$

$$\mathbf{v}^{k+1} = \mathbf{v}^k + \Delta t_{\text{int}} \sum_{i=1}^{s} (b_i \Delta \mathbf{v}^{k,i}). \tag{11b}$$

The number of stages s and parameters b_i, c_i, a_{ij} and γ_{ij} determine order, stability and numerical damping of the integration (Bursi et al. 2008; Rang 2015). They are specified in adherence to certain order conditions, refer for example to (Rang 2015) for an overview of these order conditions. With parameters $s = 1$, $b_1 = 1$, $c_1 = 0$ and $\gamma_{11} = 1$, the method has order 1 and maximum numerical damping, and actually simplifies to the linear-implicit Euler method (Arnold, Burgermeister, and Eichberger 2007).

Equation (8) is further modified to reduce the computational effort of the integration. While the mass and kinematics matrices as well as the Jacobians are already approximated by the ones from the linearization state $\mathbf{r}_{\text{lin}}$, $\mathbf{v}_{\text{lin}}$ and $\mathbf{u}_{\text{lin}}$, the EOM are still evaluated in each integration stage i to compute the generalized forces $\mathbf{h}(\tilde{\mathbf{r}}^{k,i}, \tilde{\mathbf{v}}^{k,i}, \tilde{\mathbf{u}}^{k,i})$. To prevent this time-consuming computation, the generalized forces are approximated using the right-hand side of the EOM linearization (2b). Furthermore, the linear equation system (8) is separated, leading to the linear equation system

$$\begin{aligned}\left(\mathbf{M}_{\text{lin}} - \Delta t_{\text{int}} \gamma_{ii} \mathbf{J}_{\text{v,lin}} - (\Delta t_{\text{int}} \gamma_{ii})^2 \mathbf{J}_{\text{r,lin}} \mathbf{K}_{\text{lin}}\right) \Delta \mathbf{v}^{k,i} &= \mathbf{h}_{\text{lin}} + \mathbf{J}_{\text{r,lin}} (\tilde{\mathbf{r}}^{k,i} - \mathbf{r}_{\text{lin}}) + \mathbf{J}_{\text{v,lin}} (\tilde{\mathbf{v}}^{k,i} - \mathbf{v}_{\text{lin}}) \\ &+ \mathbf{J}_{\text{u,lin}} (\tilde{\mathbf{u}}^{k,i} - \mathbf{u}_{\text{lin}}) + \Delta t_{\text{int}} \begin{bmatrix} \mathbf{J}_{\text{r,lin}} & \mathbf{J}_{\text{v,lin}} \end{bmatrix} \sum_{j=1}^{i-1} \left(\gamma_{ij} \begin{bmatrix} \Delta \mathbf{r}^{k,j} \\ \Delta \mathbf{v}^{k,j} \end{bmatrix} \right) + \Delta t_{\text{int}} \mathbf{J}_{\text{u,lin}} \sum_{j=1}^{i} \left(\gamma_{ij} \Delta \mathbf{u}^k \right) \\ &+ \Delta t_{\text{int}} \gamma_{ii} \mathbf{J}_{\text{r,lin}} \mathbf{K}_{\text{lin}} \left(\tilde{\mathbf{v}}^{k,i} + \Delta t_{\text{int}} \sum_{j=1}^{i-1} \left(\gamma_{ij} \Delta \mathbf{v}^{k,j} \right) \right)\end{aligned} \tag{12}$$

for $\Delta \mathbf{v}^{k,i}$ with half the number of unknowns compared to (8) and the equation

$$\Delta \mathbf{r}^{k,i} = \mathbf{K}_{\text{lin}} \left(\tilde{\mathbf{v}}^{k,i} + \Delta t_{\text{int}} \gamma_{ii} \Delta \mathbf{v}^{k,i} + \Delta t_{\text{int}} \sum_{j=1}^{i-1} (\gamma_{ij} \Delta \mathbf{v}^{k,j}) \right) \tag{13}$$

for direct calculation of $\Delta \mathbf{r}^{k,i}$. As a result, $\Delta \mathbf{r}^{k,i}$ and $\Delta \mathbf{v}^{k,i}$ can be obtained with less computational effort than through solving the large linear equation system (8).

There are many parameter sets for multi-stage ROW- and W-methods in literature, a few of which we considered for the comparison in this study, see Table 1. This includes the L-stable integration methods with two (LSRT2) and three (LSRT3) stages by Bursi et al. (Bursi et al. 2008). They have order 2 and 3, respectively. The parameter set for two stages, given in (Arnold,

Burgermeister, and Eichberger 2007) and in the following referred to as ABE2, provides order 3, which reduces to order 2 if the Jacobians are approximated, and features no numerical damping. The last method we considered is ROS3PRw with three stages derived by Rang (Rang 2015).

2.3. Non-iterative implicit HHT-α method

Based on the conventional implicit HHT-α method (Hilber, Hughes, and Taylor 1977), Kim et al. (Kim, Song, and Kim 2019) proposed a non-iterative implicit HHT-α integrator. The conventional method has shown good performance in multibody simulation, as summarized in (Kim, Song, and Kim 2019). Its numerical damping properties can easily be controlled through the numerical damping parameter α (Hilber, Hughes, and Taylor 1977). Based on the equations in (Kim, Song, and Kim 2019), the non-iterative implicit HHT-α method is modified for integration of EOM (1) with explicit consideration of the inputs $\mathbf{u}$ and EOM evaluation and linearization in the arbitrary linearization state $\mathbf{r}_{\text{lin}}$, $\mathbf{v}_{\text{lin}}$ and $\mathbf{u}_{\text{lin}}$ to

$$\Delta\mathbf{r}^k = \mathbf{K}_{\text{lin}}\left(\mathbf{v}^k + \frac{\Delta t_{\text{int}}}{2}\left(1 - \frac{2\beta}{\gamma}\right)\dot{\mathbf{v}}^k + \Delta t_{\text{int}}\frac{\beta}{\gamma}\Delta\mathbf{v}^k\right) \tag{14}$$

and

$$\begin{aligned}\left(\mathbf{M}_{\text{lin}} - \Delta t_{\text{int}}\gamma\tilde{\mathbf{J}}_{\text{v,lin}} - \Delta t_{\text{int}}^2\beta\tilde{\mathbf{J}}_{\text{r,lin}}\mathbf{K}_{\text{lin}}\right)\Delta\mathbf{v}^k = \mathbf{h}(\mathbf{r}^k, \mathbf{v}^k, \mathbf{u}^k) + \Delta t_{\text{int}}\gamma\tilde{\mathbf{J}}_{\text{r,lin}}\mathbf{K}_{\text{lin}} \\ \left(\mathbf{v}^k + \frac{\Delta t_{\text{int}}}{2}\left(1 - \frac{2\beta}{\gamma}\right)\dot{\mathbf{v}}^k\right) + \Delta t_{\text{int}}\gamma(1+\alpha)\tilde{\mathbf{J}}_{\text{u,lin}}\Delta\mathbf{u}^k.\end{aligned} \tag{15}$$

The vector of generalized accelerations $\dot{\mathbf{v}}^k$ is calculated with

$$\mathbf{M}_{\text{lin}}\dot{\mathbf{v}}^k = \mathbf{h}(\mathbf{r}^k, \mathbf{v}^k, \mathbf{u}^k) \tag{16}$$

and the Jacobians $\tilde{\mathbf{J}}_{\text{r,lin}}$, $\tilde{\mathbf{J}}_{\text{v,lin}}$ and $\tilde{\mathbf{J}}_{\text{u,lin}}$ are defined as

$$\tilde{\mathbf{J}}_{\text{r,lin}} = (1+\alpha)\mathbf{J}_{\text{r,lin}} \tag{17a}$$

$$\tilde{\mathbf{J}}_{\text{v,lin}} = (1+\alpha)\mathbf{J}_{\text{v,lin}} \tag{17b}$$

$$\tilde{\mathbf{J}}_{\text{u,lin}} = (1+\alpha)\mathbf{J}_{\text{u,lin}}. \tag{17c}$$

Similarly to the previously presented integration methods, we approximate the vector of generalized forces $\mathbf{h}(\mathbf{r}^k, \mathbf{v}^k, \mathbf{u}^k)$ using the right-hand side of the EOM linearization (2b), exploiting the already known vector of generalized forces and Jacobians from the linearization state.

With the solution of the linear equation system (15) for $\Delta\mathbf{v}^k$ and subsequent calculation of $\Delta\mathbf{r}^k$ with (14), the generalized coordinates and velocities in integration step $k+1$ are calculated as

$$\mathbf{r}^{k+1} = \mathbf{r}^k + \Delta t_{\text{int}}\Delta\mathbf{r}^k \tag{18a}$$

$$\mathbf{v}^{k+1} = \mathbf{v}^k + \Delta t_{\text{int}}\Delta\mathbf{v}^k. \tag{18b}$$

α, β and γ are parameters of the method. The last two can be expressed in terms of the numerical damping parameter α as

$$\beta = \frac{(1-\alpha)^2}{4} \tag{19a}$$

$$\gamma = \frac{1-2\alpha}{2}. \tag{19b}$$

A feasible range for the numerical damping parameter is $-1/3 \leq \alpha \leq 0$, with zero numerical damping for $\alpha = 0$ and increasing numerical damping for decreasing α (Hilber, Hughes, and Taylor 1977). We chose three different values of α for the following analysis, see Table 1: The first one is $\alpha = 0$, i.e., no numerical damping; the second one is $\alpha = -1/3$ with maximum numerical damping in the given parameter range; the third one is $\alpha = -0.1$, which was also chosen in (Hilber, Hughes, and Taylor 1977) for a comparison with other dissipative integration algorithms. It should be noted that the method with $\alpha = 0$ equals the ROW-/W-method with parameter values $s = 1$, $b_1 = 1$, $c_1 = 0$ and $\gamma_{11} = 0.5$.

2.4. Computational workflow of the EOM linearization and integration

The presented non-iterative implicit integration methods have been simplified by exploiting the EOM linearization (2) in the arbitrary linearization state $\mathbf{r}_{\text{lin}}$, $\mathbf{v}_{\text{lin}}$ and $\mathbf{u}_{\text{lin}}$. This linearization state can either be chosen to be the state in the current integration step k, the state in a previous integration step, or even the initial configuration before real-time simulation. In the latter cases, not only the Jacobians are approximated with those from the linearization state, but also the EOM themselves are approximated based on the linearization. Thus, the demanding EOM evaluation and Jacobian computation in each integration step are avoided.

Figure 1 shows the computational workflow of the EOM linearization and integration. After the initialization and an initial EOM evaluation and linearization, the generalized coordinates and velocities are computed in each integration step k with one of the non-iterative implicit integration methods. The simulation ends as soon as the specified simulation time t_S is reached. Updating the EOM evaluation and linearization is necessary when the generalized coordinates or velocities change significantly, causing significant change of the mass matrix, kinematics matrix and, in particular, the Jacobians. Theoretically, the EOM evaluation and linearization could be updated adaptively, meaning more frequent updates when the system configuration changes a lot and less frequent updates when it changes only slightly. Real-time simulation, however, requires deterministic computational effort to ensure that the real-time requirement is met in each single time step. To this end, the EOM evaluation and linearization is updated at a constant frequency, regardless of the actual configuration change. To specify this frequency, a fixed linearization step

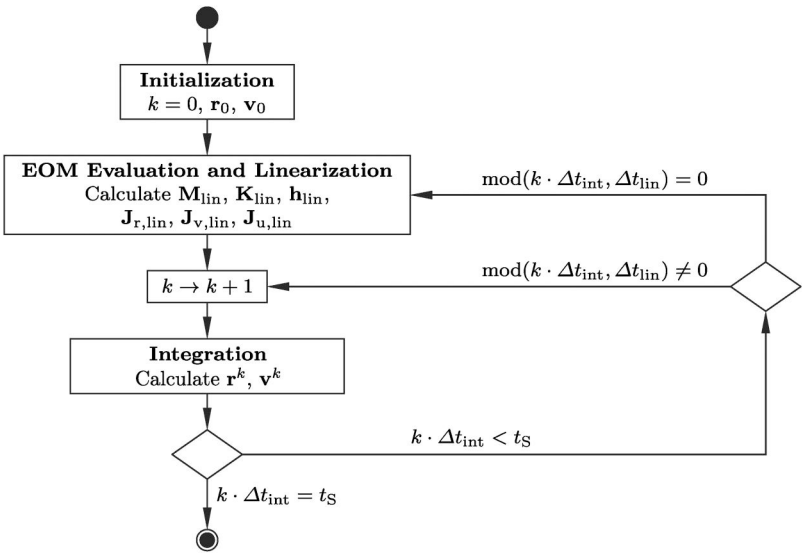


Figure 1. Flow chart of the computational workflow.

size Δt_{lin} is introduced that is an integer multiple of the integration step size Δt_{int}. Depending on its choice, the EOM evaluation and linearization is updated before the next integration step, or the next integration step is directly performed reusing the previous EOM evaluation and linearization.

Updating the EOM evaluation and linearization only every few time steps saves computation time over the entire simulation time t_S. However, the computation time in the different simulation steps is not constant. Simulation steps that involve the computationally demanding EOM evaluation and linearization require more computational effort than those that consist solely of integration steps. In contrast, real-time applications usually run with a constant step size Δt_{RT}, providing the same time span for completing all required computations in each time step. There are essentially two ways for exploiting the computational workflow depicted in Fig. 1 in such a real-time application. The first option is to choose the integration step size as $\Delta t_{\text{int}} = \Delta t_{\text{RT}}$. In this case, one integration step is performed in each time step of the real-time simulation, while the EOM evaluation and linearization is updated only every few time steps. The real-time capability of the simulation is then determined by the computationally most demanding time steps that include the EOM evaluation and linearization update. In the remaining time steps, the computational effort is reduced, allowing the available calculation time to be used for external computations that may run simultaneously in the real-time application. The second option is to choose the linearization step size as $\Delta t_{\text{lin}} = \Delta t_{\text{RT}}$. In this case, each time step of the real-time simulation includes one EOM evaluation and linearization update as well as several integration steps. This can potentially increase simulation accuracy with only a slight increase in computational effort, compared to a real-time simulation with an EOM evaluation and linearization update and only one integration step per time step.

3. Multibody system of the double wishbone suspension

The considered double wishbone suspension is a representative example for passenger vehicles in terms of topology and parametrization. It is a challenging problem with respect to real-time simulation due to the high number of model elements and degrees of freedom as well as its numerical stiffness.

3.1. Topology of the suspension

The suspension system features the typical topology of a double wishbone suspension, as shown Fig. 2. The wheel mounts are connected to the vehicle body via upper wishbones, lower wishbones and trailing arms. Furthermore, they are connected to the steering rack via track rods. The damper

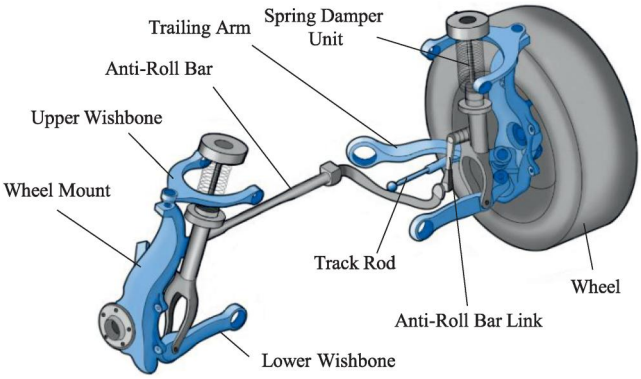


Figure 2. Double wishbone suspension.

in each spring damper unit is modeled with a damper tube and a piston rod, connected by a point-to-point force element with nonlinear damping characteristics. Similarly, the main springs are each separated into two bodies with a point-to-point force element in between representing the spring characteristics. Both sides of the suspension are coupled through an anti-roll bar, consisting of two rigid bodies and a discrete rotational spring. The anti-roll bar is connected to the damper tubes via anti-roll bar links. Furthermore, the wheels are connected to the wheel mounts with bushing force elements representing the wheel bearing stiffness. All bodies are considered as rigid. They are connected to each other through bushing force elements with generally nonlinear stiffness and damping properties in all three translational and rotational directions. Most of the connections in the suspension system are actual elastic bushings with functional stiffness and damping properties to represent the elastokinematic properties of the suspension. The few kinematic joints in the system are replaced by bushing force elements with appropriate stiffness and damping values in order to avoid kinematic constraints. As already mentioned, point-to-point force elements are used to model the damper and main spring characteristics. Furthermore, the wheel travel is limited by bump and rebound stops with highly progressive stiffness.

The suspension model provides all inputs that are necessary for integration into a full vehicle model. The motion of the vehicle body can be imposed as a prescribed motion. In the same way, a steering input can be imposed as a prescribed motion of the steering rack. Furthermore, tire forces and torques as well as drive and brake torques can be applied onto the wheels.

All in all, the multibody system of the suspension consists of 24 rigid bodies (not taking into account the vehicle body and the steering rack whose motions are prescribed) and 55 force elements, resulting in a total of 144 degrees of freedom.

3.2. Equations of motion

All bodies in the multibody system of the suspension are solely connected through force elements, without using any kinematic constraint. Thus, its dynamics is directly described by the nonlinear ordinary differential EOM (1), which is obtained by combining all Newton-Euler equations of the separate bodies. Here, the vectors of generalized coordinates and velocities of a body l are defined as

$$\mathbf{r}_l = \begin{bmatrix} {}^0x_l & {}^0y_l & {}^0z_l & \alpha_{z,l} & \alpha_{y',l} & \alpha_{x'',l} \end{bmatrix}^T \tag{20a}$$

$$\mathbf{v}_l = \begin{bmatrix} {}^0v_{x,l} & {}^0v_{y,l} & {}^0v_{z,l} & {}^B\omega_{x,l} & {}^B\omega_{y,l} & {}^B\omega_{z,l} \end{bmatrix}^T \tag{20b}$$

and combined to the generalized vectors of the overall system as

$$\mathbf{r} = \begin{bmatrix} \mathbf{r}_1^T & \cdots & \mathbf{r}_m^T \end{bmatrix}^T \tag{21a}$$

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_1^T & \cdots & \mathbf{v}_m^T \end{bmatrix}^T, \tag{21b}$$

where m is the total number of bodies. ${}^0x_l$, ${}^0y_l$, ${}^0z_l$ and ${}^0v_{x,l}$, ${}^0v_{y,l}$, ${}^0v_{z,l}$ denote the displacement of the body w.r.t to the design configuration and its velocity in the global coordinate system. The rotational kinematics is described by yaw-pitch-roll-angles $\alpha_{z,l}$, $\alpha_{y',l}$ and $\alpha_{x'',l}$ with a current frame rotation sequence, and angular velocities ${}^B\omega_{x,l}$, ${}^B\omega_{y,l}$, ${}^B\omega_{z,l}$ given in the corresponding body coordinate system. Singularities are avoided by exploiting the limited motion range of the bodies and defining their body coordinate systems accordingly to prevent large rotations about the critical axis. Because of the parametrization of the suspension model, the EOM have a high numerical stiffness with a stiffness ratio SR $> 5 \cdot 10^4$ in the design configuration.

3.3. Implementation

The model of the double wishbone suspension system is implemented in Matlab/Simulink R2020b. It is divided mainly into two subsystems, namely the EOM evaluation and linearization subsystem as well as the integration subsystem according to Fig. 1. Each subsystem consists of several Matlab function blocks, in which the model equations of the different multibody components and the integration formulas are implemented.

In the EOM evaluation and linearization subsystem, all model elements of the suspension are calculated to formulate the EOM. The linearization is performed analytically in order to adhere to the time restrictions of real-time simulations. The analytical linearization procedure consists of calculating all required partial derivatives of the model elements and combining them appropriately to set up the Jacobians. More details on this procedure can be found in (Archut and Corves, 2025).

The integration of the EOM is performed in the integration subsystem. All integration methods from Sect. 2 have in common that rather large linear equation systems must be solved. To this end, the corresponding coefficient matrices are decomposed using LU decomposition and the linear equation systems are solved through forward and backward substitution. Since the coefficient matrices only change if the EOM evaluation and linearization is updated, LU decomposition is performed only once after each update.

Based on the Matlab/Simulink model, C Code is generated with the Simulink Coder (MathWorks, 2025). The Intel oneAPI Math Kernel Library (Intel) is employed in the code generation process. This library contains optimized implementations of the Basic Linear Algebra Subprograms (BLAS) and the Linear Algebra Package (LAPACK). During compilation, Simulink Coder generates calls to BLAS and LAPACK functions to speed up linear algebra operations.

4. Dynamic suspension tests

Numerical simulations of dynamic suspension tests with the double wishbone suspension model were performed to analyze the performance of the integration methods. To this end, inputs to the model were defined, the model was simulated using the different integration methods, and error quantities relative to reference solutions were calculated. In the following, the dynamic suspension tests are described.

4.1. Load cases

Four different load cases were considered for the analysis, categorized into *full vehicle load cases* and *test rig load cases*. Full vehicle load cases were used to simulate extreme driving conditions. To obtain the model inputs for these load cases, the suspension model was incorporated as a subsystem into a nonlinear full vehicle two-track model, which was completed by a Magic Formula tire model. Driving maneuvers were performed, and all inputs to the suspension subsystem were recorded. The recorded inputs were then utilized to perform dynamic suspension tests with the suspension model alone, employing the different integration methods. The following driving maneuvers, which focus on the lateral dynamics, were considered.

Sine with dwell test: The sine with dwell test, typically used to evaluate electronic stability control systems, is standardized according to ISO 19365 for validating vehicle dynamics simulation (ISO 19365:2016 19365:2016, 2016). In this test, the vehicle is coasting from an initial velocity of 80 km/h. The steering input consists of a sine wave with a frequency of 0.7 Hz and a dwell time of 500 ms at the second peak amplitude, see Fig. 3. The maximum steering angle is 2.5 $\delta_{0.3g}$, with $\delta_{0.3g}$ being the angle that produces a lateral acceleration of 0.3 g in a slowly increasing steer test at 80 km/h. The maximum lateral acceleration in the sine with dwell test is close to 0.7 g.

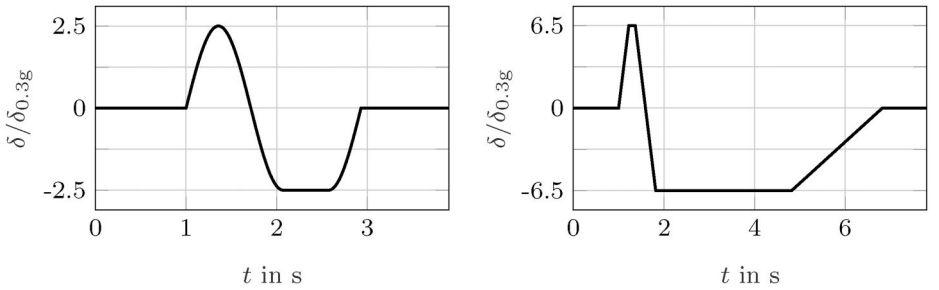


Figure 3. Steering inputs δ for the sine with dwell test (left) and the fishhook maneuver (right).

Fishhook maneuver: The fishhook maneuver is a standardized maneuver designed to maximize the roll motion of a vehicle and is usually used as an anti-rollover test (U.S. Department of Transportation and National Highway Traffic Safety Administration 2013). The steering input is shown in Fig. 3 and consists of a first steer with a maximum steering angle of 6.5 $\delta_{0.3g}$, a short dwell time, and then a counter steer with the same maximum steering angle. After a second dwell time, the steering angle is returned to zero. During the maneuver, the vehicle is coasting from an initial velocity of 50 mph, which is approximately 80 km/h. The maximum lateral acceleration is close to 1.1 g with a momentary liftoff of the right front wheel.

The test rig load cases are not standardized but designed to find the accuracy limits of the integration methods for extreme conditions, which may be worse than usual driving conditions. In these load cases, the vehicle body is fixed and forces are applied on the wheel center points of both wheels. All other inputs, namely steering, wheel torques, drive torques and brake torques, are zero. The following test rig load cases, which focus on the vertical dynamics of the suspension, were considered. The given forces are specified in the vehicle body coordinate system, with the x-axis pointing to the front, the y-axis pointing to the left and the z-axis pointing upwards.

Force step: In this load case, a force step is applied with a step height of $\widetilde{\mathbf{f}}_{\mathrm{WCP}} = (\,0,\,0,\,15000\,)^T$ N. Before the force step input, the suspension is in static equilibrium at a force $\mathbf{f}_{0,\mathrm{WCP}} = (\,0,\,0,\,5000\,)^T$ N. This load case produces a large wheel travel and activates the highly stiff and nonlinear bump stops of the suspension.

Force frequency sweep: Starting from the static equilibrium at a constant force $\mathbf{f}_{0,\mathrm{WCP}} = (\,0,\,3000,\,5000\,)^T$ N, a force frequency sweep from 0 Hz to 30 Hz with a constant amplitude of $\widehat{\mathbf{f}}_{\mathrm{WCP}} = (\,-1500,\,1500,\,3000\,)^T$ N and a duration of 10 s is applied. This excites the suspension with high longitudinal and lateral forces at different frequencies.

4.2. Experimental design and evaluation of the numerical simulations

The dynamic suspension tests were performed with the integration methods shown in Table 1 and integration and linearization step sizes Δt_{int} and Δt_{lin} ranging between 0.1 ms and 10 ms.

The accuracy of the non-iterative implicit integration methods was evaluated through comparisons with reference results. The Matlab/Simulink model was recently compared to a model of the same suspension in the commercial multibody software Simpack (Archut and Corves, 2025). In the present analysis, we want to focus on the integration methods themselves and compare them to a solver for numerically stiff systems delivered with Matlab/Simulink. Therefore, the reference results were obtained with the Matlab/Simulink model using the solver ode23tb with an error tolerance of 10^{-4}. This solver employs an implicit Runge-Kutta formula with a trapezoidal rule step as first stage and a backward differentiation formula as second stage (MathWorks, 2025). The deviations between the simulation results obtained with the non-iterative implicit integration methods and the reference results were quantified with the root mean square error defined as

$$e_y = \sqrt{\left(\frac{\sum_{i=1}^{n}(y_i - y_{\mathrm{i,ref}})^2}{n}\right)}. \tag{22}$$

Here, y_i and $y_{\mathrm{i,ref}}$ are the values of a signal and the reference signal at the sample i, respectively, and n is the total number of samples. The evaluated signals for the full vehicle load cases are the lateral wheel displacement $\mathbf{s}_{\mathrm{WCP,y}}$ of the left wheel center point and the total lateral force $\mathbf{f}_{\mathrm{VB,y}}$ that is applied from the suspension system onto the vehicle body. Both quantities are given in the vehicle body coordinate system. For the test rig load cases, the same quantities were considered but in vertical direction, i.e., $\mathbf{s}_{\mathrm{WCP,z}}$ and $\mathbf{f}_{\mathrm{VB,z}}$, since these load cases focus more on the vertical than the lateral dynamics.

5. Results

In this section, the results of the suspension tests are analyzed. This includes the accuracy with EOM evaluation and linearization update before each integration step and, subsequently, only every few integration steps. Before that, computation times are presented to evaluate the accuracy in relation to the computational effort.

5.1. Computation times

The computation times are evaluated using the real-time factor

$$\mathrm{RTF} = t_C / t_S, \tag{23}$$

which is the ratio between the specified simulation time t_S and the computation time t_C required to complete the simulation. For $\mathrm{RTF} \leq 1$, the simulation is considered as real-time capable. Please note that definition (23) evaluates real-time capability as an average over the entire simulation time t_S, regardless of the actual computational effort in the individual time steps, as discussed in Sect. 2.4. The real-time factor was determined as a mean value of multiple simulations with a simulation time of $t_S = 10$ s on a personal computer with an Intel Core i7-3770 CPU at 3.40 GHz. The results for the linear-implicit Euler method, the ROW-/W-method with different number of stages and the non-iterative implicit HHT-α method are shown in Fig. 4.

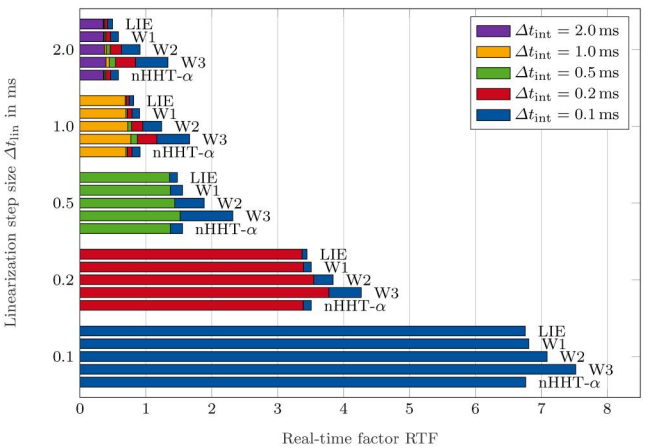


Figure 4. Real-time factor RTF of the integration methods for different linearization step sizes Δt_{lin} and integration step sizes Δt_{int}.

The results show that, independent of the integration method, the largest amount of the computation time is required for the EOM evaluation and linearization and subsequent LU decomposition of the corresponding coefficient matrix, of which one execution takes approximately 0.68 ms in total. Accordingly, the real-time factor increases significantly with decreasing linearization step size, and real-time capability can only be achieved with linearization step sizes Δt_{lin} down to 1 ms. In contrast, the actual integration of the generalized coordinates and velocities is computationally less demanding. A single integration step with the linear-implicit Euler method requires only approximately 0.014 ms. Both the nHHT-α method and the ROW-/W-method with one stage (W1) require nearly identical computation times, approximately 1.6 times that of the linear-implicit Euler method. The computation times of the ROW-/W-method increase with the number of stages: by a factor of 2.6 for two stages (W2) and by a factor of 4.5 for three stages (W3) compared to that for one stage. Given the low computation times of an integration step compared to an EOM evaluation and linearization update, the choice of the integration method has only a moderate impact on the real-time factor. For instance, using the ROW-/W-method with three stages increases the real-time factor by only about 11 % compared to the linear-implicit Euler method, when the EOM evaluation and linearization is updated before each integration step.

Furthermore, performing multiple integration steps between two EOM evaluation and linearization updates, i.e., $\Delta t_{\text{int}} < \Delta t_{\text{lin}}$, does not drastically increase the real-time factor. As an example, halving the integration step size while maintaining the same linearization step size leads to an increase of only 2 % when using the linear-implicit Euler method, and an increase of 13 % with the ROW-/W-method with three stages. Conversely, doubling the linearization step size while keeping the integration step size constant nearly halves the real-time factor, with a decrease of 49 % for the linear-implicit Euler method and 44 % for the ROW-/W-method with three stages. These results demonstrate the good scalability of the required computation times through the choice of the integration and linearization step sizes.

All in all, real-time capability is achieved on the considered hardware with a minimum linearization step size of $\Delta t_{\text{lin}} = 1$ ms. The integration step size Δt_{int} can be selected smaller, depending on the integration method.

5.2. Accuracy with EOM evaluation and linearization update before each integration step

The accuracy analysis begins with the results obtained with updating the EOM evaluation and linearization before each integration step, i.e., $\Delta t_{\text{lin}} = \Delta t_{\text{int}}$. The error values for the different load cases are shown in Fig. 5, with the definition of the integration methods given in Table 1. Generally, the results obtained with the linear-implicit Euler method have the lowest level of accuracy. Particularly in the test rig load cases, the error values are almost one order of magnitude higher compared to the most accurate results obtained with the other integration methods, at some step sizes even more. Only the lateral wheel displacement $e_{\text{s,y}}$ in the full vehicle load cases is computed with an accuracy similar to that of the most accurate methods. The LSRT2 method, one of the ROW-/W-methods with two stages, provides the best results, achieving the lowest error values for almost all load cases and step sizes. It outperforms the other ROW-/W-method with two stages, ABE2, and even both three-stage ROW-/W-methods, LSRT3 and ROS3PRw, despite their higher computational effort.

The accuracy of the results obtained with the non-iterative implicit HHT-α method ranks between those of the linear-implicit Euler method and the LSRT2 method. The numerical damping factor α shows a significant impact, as the error values obtained with $\alpha = 0$, i.e., without numerical damping, are up to half an order of magnitude smaller than those obtained with the other choices of α. In some cases, the accuracy with $\alpha = 0$ is even very similar to that of the LSRT2 method, provided the step size is small enough. Besides accuracy, the performance

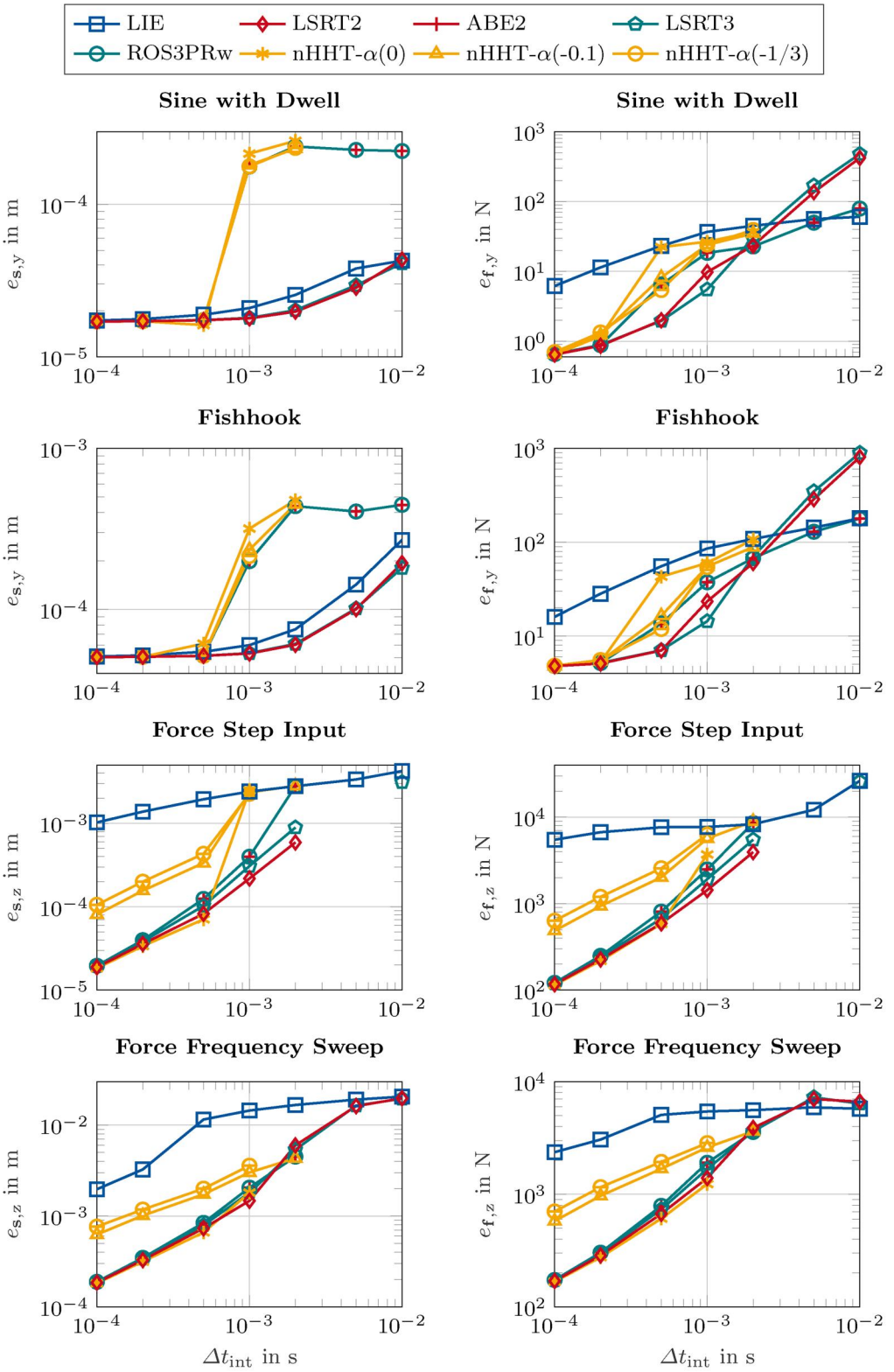


Figure 5. Root mean square errors $e_{s,y}$ and $e_{s,z}$ of the wheel displacement and $e_{f,y}$ and $e_{f,z}$ of the force on the vehicle body for the different load cases ($\Delta t_{lin} = \Delta t_{int}$).

of the integration methods also differs in terms of robustness. The methods with high numerical damping, especially the linear-implicit Euler method and the ROW-/W-methods LSRT2 and LSRT3, tend to have better robustness and provide stable simulation even with large step sizes of up to 10 ms. In contrast, simulations with the other integration methods become unstable for step sizes higher than 2 ms or even 1 ms, resulting in infinite error values not plotted in Fig. 5.

In the following, time plots obtained with step sizes of 1 ms and 0.5 ms are analyzed, as these step sizes are particularly relevant for vehicle dynamics simulation. The integration methods selected for this analysis include the linear-implicit Euler method, since it is often used for real-time vehicle dynamics simulation; the LSRT2 method, since it provides the best results across the considered load cases; and the non-iterative implicit HHT-α method with $\alpha = 0$, since it requires slightly less computational effort than the LSRT2 method but achieves comparable results in some cases.

Figure 6 shows the results for the sine with dwell test. The time histories of the wheel displacement and the vehicle body force confirm the high accuracy of the LSRT2 method. Even with a step size of 1 ms, deviations from the reference results remain within the range of plot accuracy. In contrast, the linear-implicit Euler method fails to accurately capture the oscillations of the vehicle body force caused by the sudden stop of the steering motion at $t \approx 3$ s, as these oscillations are damped out too quickly. The solutions obtained with the non-iterative implicit HHT-α method depend heavily on the step size. While the results obtained with a step size of 0.5 ms are within the range of plot accuracy, the results with a step size of 1 ms exhibit an offset of the

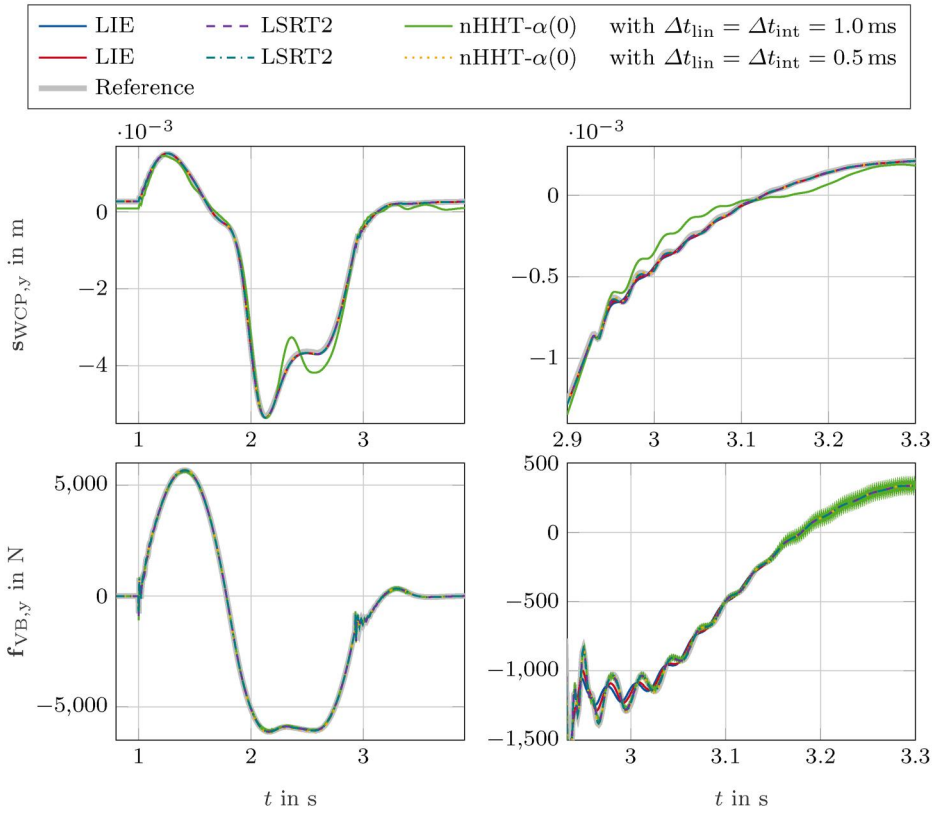


Figure 6. Lateral wheel displacement $\mathbf{s}_{\mathrm{WCP,y}}$ and lateral force $\mathbf{f}_{\mathrm{VB,y}}$ on the vehicle body for the sine with dwell test.

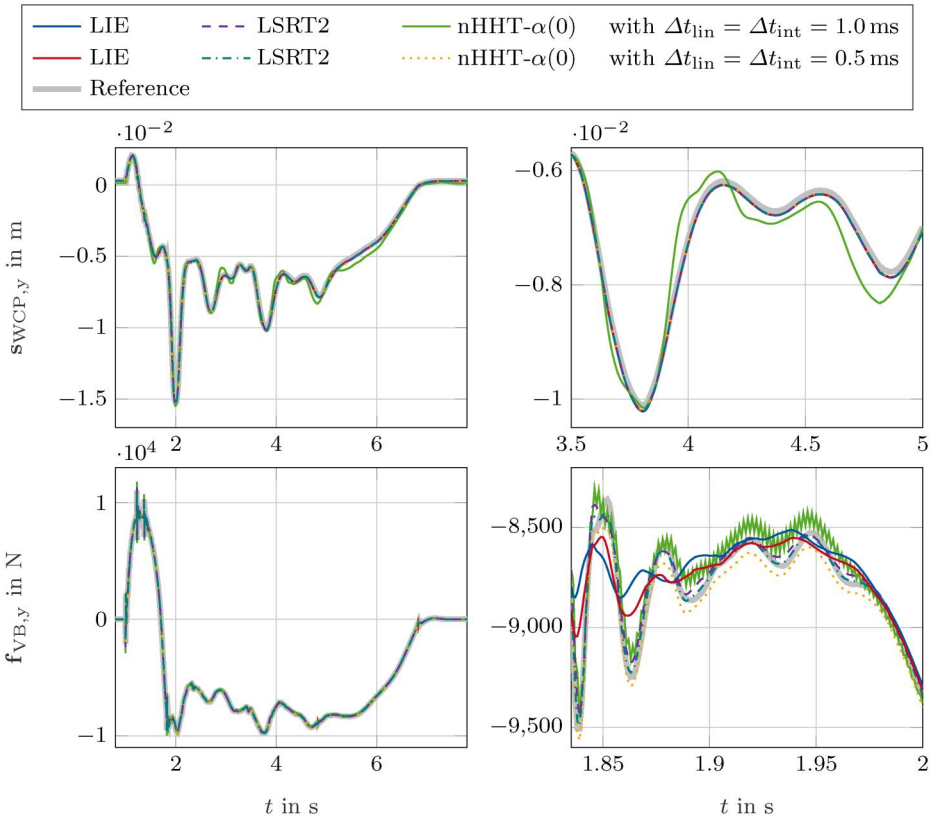


Figure 7. Lateral wheel displacement $\mathbf{s}_{WCP,y}$ and lateral force $\mathbf{f}_{VB,y}$ on the vehicle body for the fishhook maneuver.

wheel displacement in the static equilibrium before the sine with dwell maneuver as well as deviations during the maneuver. Moreover, unphysical oscillations of the vehicle body force occur at $t \approx 3.2$ s.

The results for the fishhook maneuver, presented in Fig. 7, show similar characteristics to those observed for the sine with dwell test. However, this maneuver is slightly more demanding. Consequently, the time histories of the vehicle body force obtained using the LSRT2 method with a step size of 1 ms and the non-iterative implicit HHT-α method with a step size of 0.5 ms have slightly higher deviations from the reference results compared to those in the sine with dwell test. The LSRT2 method with a step size of 0.5 ms still provides very high accuracy, with deviations remaining within the range of plot accuracy.

Figure 8 shows the results for the force step input. Again, the time histories obtained with the LSRT2 method show very high accuracy, even with a step size of 1 ms. The results obtained with the linear-implicit Euler method reveal excessive damping, causing the oscillations to decay too quickly as already observed for the full vehicle load cases. Furthermore, an offset of the wheel displacement in the static equilibrium before and deviations in the oscillations after the force step input are observed in the results obtained with the non-iterative implicit HHT-α method with a step size of 1 ms.

The results for the force frequency sweep are given in Fig. 9. The time histories obtained with the LSRT2 method and the non-iterative implicit HHT-α method are very similar. Highly accurate results are achieved with a step size of 0.5 ms, with deviations remaining within the range of plot accuracy throughout the entire frequency range. With a step size of 1 ms, slightly higher deviations can be observed at higher frequencies, specifically in the time interval 8 s $\leq t \leq$ 8.2 s.

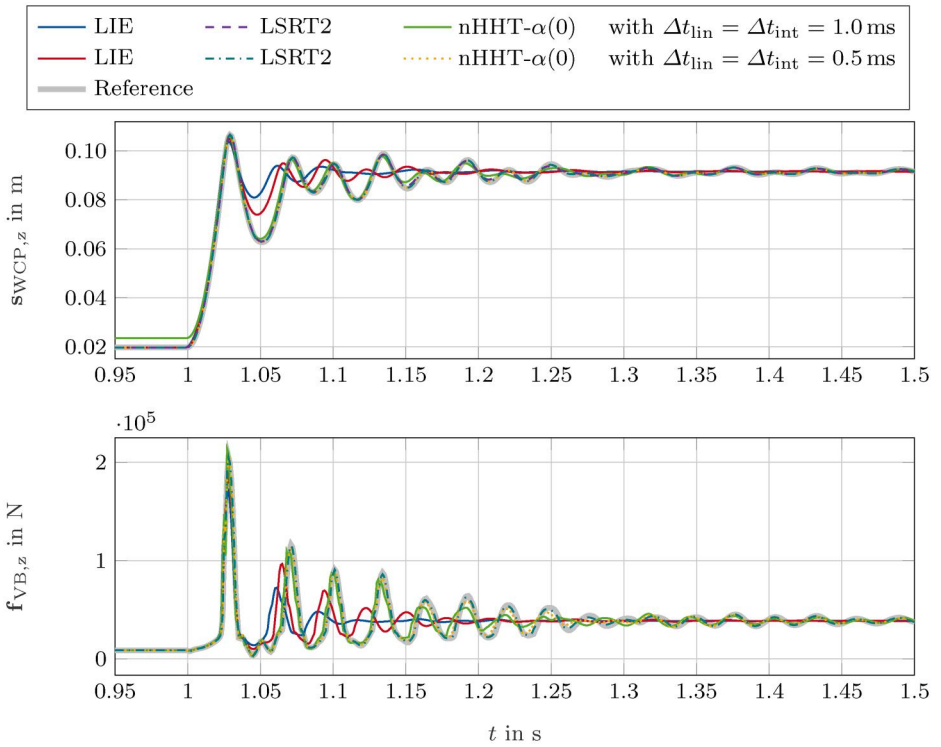


Figure 8. Vertical wheel displacement $\mathbf{s}_{\mathrm{WCP},z}$ and vertical force $\mathbf{f}_{\mathrm{VB},z}$ on the vehicle body for the force step input.

The linear-implicit Euler method, once again, fails to accurately represent the system response, not only at high but also at low frequencies in the time interval $3\ \mathrm{s} \leq t \leq 3.5\ \mathrm{s}$.

Overall, the LSRT2 method demonstrates the best performance in terms of accuracy, with deviations remaining within the range of plot accuracy even with a step size of 1 ms in most of the cases. The non-iterative implicit HHT-α method can provide very similar accuracy but requires a step size of 0.5 ms. With this step size, however, computation times are almost twice as high as those of the LSRT2 method with a step size of 1 ms, meaning that real-time capability can no longer be maintained, cf. Fig. 4. With a step size of 1 ms, the method does not provide satisfactory accuracy, as deviations of the wheel displacement and unphysical oscillations of the vehicle body force appear. Furthermore, the results indicate that the linear-implicit Euler method, despite requiring the lowest computational effort, is not well suited for this complex application. The error values are in some cases more than one order of magnitude higher than those obtained with the other integration methods, and the time histories reveal excessive damping causing oscillations to decay too quickly.

5.3. Accuracy with EOM evaluation and linearization updates every few integration steps

The analysis in Sect. 5.1 demonstrated the scalability of the required computation times through different choices of the integration and linearization step sizes Δt_{int} and Δt_{lin}. This raises the following questions regarding simulation accuracy and computation times: Is it possible to improve the accuracy achieved with a specific integration method by reducing the integration step size while maintaining the same linearization step size? This approach would involve performing multiple integration steps reusing the EOM evaluation and linearization from a previous step. Additionally, is it possible by reducing the integration step size to achieve a better compromise

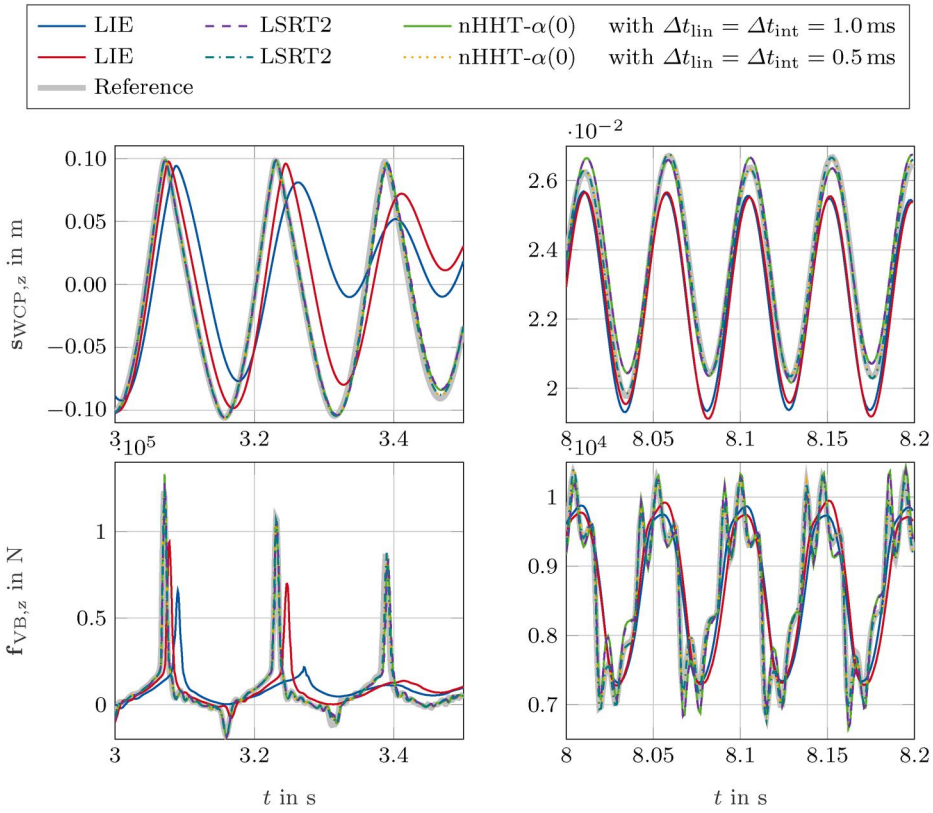


Figure 9. Vertical wheel displacement $\mathbf{s}_{\mathrm{WCP},z}$ and vertical force $\mathbf{f}_{\mathrm{VB},z}$ on the vehicle body for the force frequency sweep.

between simulation accuracy and computation time with one integration method than with another integration method and an EOM evaluation and linearization update before each integration step? To address these questions, Fig 10 presents the error values obtained with different integration step sizes Δt_{int} and fixed linearization step size Δt_{lin}. The following conclusions can be drawn from these results.

Simulation accuracy is strongly affected by the choice of the linearization step size. As the linearization step size decreases, i.e., the EOM evaluation and linearization is updated more frequently, the resulting error values also decrease. This applies to all integration methods but becomes particularly evident in the results obtained with the LSRT2 method. Here, reductions of up to one order of magnitude are observed for the full vehicle load cases, and up to half an order of magnitude for the test rig load cases, when the linearization step size is reduced from 2 ms to 1 ms. A further reduction from 1 ms to 0.5 ms leads to a slightly smaller improvement of the error values, especially for the full vehicle load cases. These results indicate the high nonlinearity of the suspension model, which necessitates frequent updates of the EOM evaluation and linearization.

Furthermore, simulation accuracy can be improved for a given linearization step size by reducing the integration step size. However, the extent of this effect varies across the different integration methods. The error values obtained with the linear-implicit Euler method show a significant reduction. As an example, with a linearization step size of 1 ms, the error values of the vehicle body force in the full vehicle load cases are reduced by up to 40 % when the integration step size is halved, and in the test rig load cases, the error values of the wheel displacement are reduced by up to 20 % and those of the vehicle body force by almost 35 % and up to 55 %, depending on

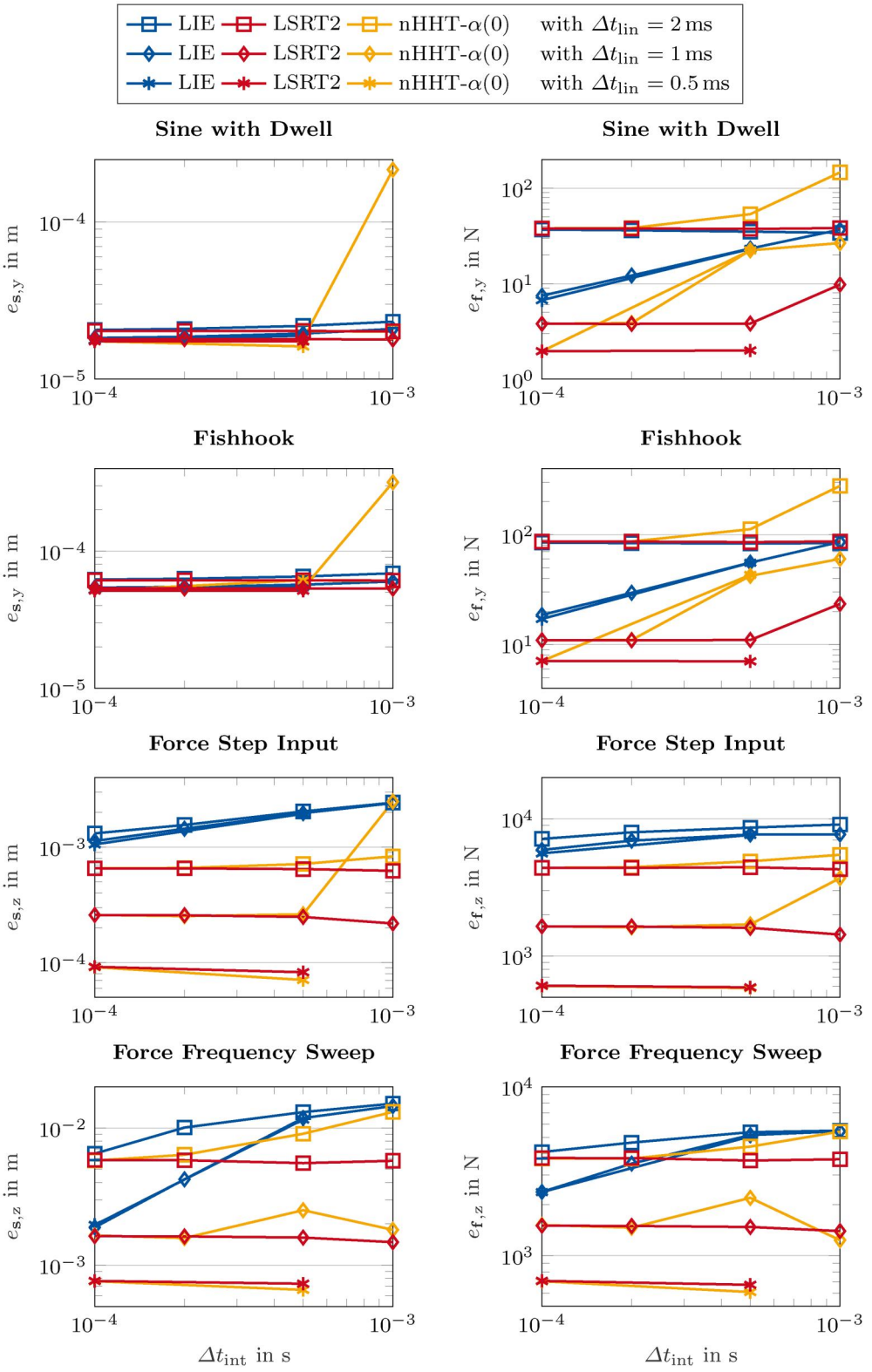


Figure 10. Root mean square errors $e_{s,y}$ and $e_{s,z}$ of the wheel displacement and $e_{f,y}$ and $e_{f,z}$ of the force on the vehicle body for the different load cases ($\Delta t_{lin} \geq \Delta t_{int}$).

the load case. For the non-iterative implicit HHT-α method, improved accuracy can be observed especially for the full vehicle load cases. In particular, the outliers with large error values that are associated with deviations in the wheel displacement and unphysical oscillations in the vehicle body force for $\Delta t_{\text{int}} = \Delta t_{\text{lin}} = 1$ ms (cf. Figs. 6–9) are eliminated by reducing the integration step size to 0.5 ms. For the LSRT2 method, however, no substantial accuracy improvement can be achieved by reducing the integration step size for a given linearization step size. This demonstrates that this method achieves high accuracy even with relatively large integration step sizes.

In the analysis of the simulation results with EOM evaluation and linearization update before each integration step, i.e., with $\Delta t_{\text{int}} = \Delta t_{\text{lin}}$, the LSRT2 method with a step size of 1 ms provided the best compromise between accuracy and real-time performance. No better compromise could be found with any of the other integration methods by reducing the integration step size while maintaining a constant linearization step size. Although the accuracy obtained with the linear-implicit Euler method is improved, error values remain higher than those of the LSRT2 method with a step size of 1 ms. The non-iterative implicit HHT-α method can provide comparable accuracy by reducing the integration step size, but this requires similar or even slightly higher computation times than the LSRT2 method with a step size of 1 ms. Consequently, the LSRT2 method with $\Delta t_{\text{int}} = \Delta t_{\text{lin}} = 1$ ms remains the best choice for balancing real-time capability and accuracy in this analysis.

6. Conclusion

This study simplifies various non-iterative implicit integration methods by exploiting a linearization-based EOM approximation and assesses their performance for real-time vehicle dynamics simulation, specifically focusing on multibody simulations of a double wishbone suspension. The suspension model incorporates elastic bushings to account for its elastokinematic effects on driving dynamics. The high stiffness and damping properties of the elastic bushings contribute to high numerical stiffness of the EOM, posing challenges in real-time simulations for applications such as driving simulators. Several non-iterative implicit integration methods from literature were employed to achieve real-time capability: the linear-implicit Euler method, ROW-/W-methods with different number of stages and parameters, and a non-iterative implicit HHT-α method with different amounts of numerical damping. These methods have been simplified to further reduce the computational effort, approximating the EOM and their Jacobians through analytical linearization that is updated either before each or only every few integration steps.

The analysis revealed that most of the computational effort is associated with the EOM evaluation and linearization, specifically at least 87 % when the EOM is evaluated and linearized before each integration step, depending on the integration method. Consequently, overall computation times of the various integration methods differ only moderately. Moreover, the results demonstrate the good scalability of computation times as they can be significantly reduced through less frequent EOM evaluation and linearization updates. Alternatively, the number of integration steps can be increased with only moderate increase of computation times.

Suspension tests with different full vehicle and test rig load cases were performed to analyze the accuracy of each integration method. Although frequently employed for real-time vehicle dynamics simulation, the linear-implicit Euler method did not provide satisfactory accuracy in this complex application, with error values in some cases higher by more than one order of magnitude than those of the best integration methods with the same step sizes. The ROW-/W-methods yielded the best results, with the accuracy depending on the number of stages and parameters. Notably, the analyzed three-stage methods did not provide better performance than those with two stages, despite their higher computational costs. The accuracy of the non-iterative implicit HHT-α method was between that of the linear-implicit Euler and the ROW-/W-methods.

The best results were obtained without numerical damping, achieving accuracy comparable to the ROW-/W-methods in some cases.

Furthermore, it was demonstrated that the EOM evaluation and linearization update frequency strongly affects simulation accuracy, due to the nonlinearity of the suspension model. Error values were reduced by up to one order of magnitude through a reduction of the linearization step size from 2 ms to 1 ms. The effect of reducing the integration step size for a given linearization step size depends on the integration method. While this approach was effective for the linear-implicit Euler method and the non-iterative implicit HHT-α method, the accuracy of the LSRT2 method (one of the analyzed ROW-/W-method with two stages) could not be further improved.

All in all, the LSRT2 method with a step size of 1 ms and EOM evaluation and linearization update before each integration step offered the best compromise between accuracy and real-time capability in this analysis.

The present study utilized a double wishbone suspension model as a representative example for passenger vehicles, demonstrating the effectiveness and robustness of the proposed simplifications. It should be noted that the same multibody modeling methodology and integration with the LSRT2 method have also been applied to another suspension type, namely a multilink suspension (see (Archut and Corves, 2025)). To further validate the transferability of the results, the detailed analysis and comparison of the different integration methods presented here should be extended to the multilink suspension or other suspension types, or even to multibody models of completely different systems.

Disclosure statement

The authors report there are no competing interests to declare.

AI tool disclosure statement

Artificial intelligence tools (specifically, ChatGPT by OpenAI and RWTHgpt by RWTH Aachen University) were used to assist in the refinement of English grammar, sentence structure, and clarity during the manuscript preparation process.

Data availability statement

Data will be made available on request.

References

Adamski, D. 2014. “Simulation in Der Fahrwerktechnik: Einführung in Die Erstellung Von Komponenten- Und Gesamtfahrzeugmodellen.” *ATZ / MTZ-Fachbuch*. Wiesbaden: Springer. https://doi.org/10.1007/978-3-658-06536-2

Archut, J. L., and B. Corves. 2025. “Real-Time Multibody Simulation of Vehicle Wheel Suspensions of Different Topologies with Elastokinematic Properties.” *Multibody System Dynamics* 63 (1-2): 255–271. https://doi.org/10.1007/s11044-024-09984-2

Archut, J. L., and B. Corves. 2025. “Systematic Mapping of Methods for Real-Time Capable Multibody Simulation of Road Vehicles Using PRISMA.” *Multibody System Dynamics*. https://doi.org/10.1007/s11044-025-10081-1

Arnold, M., B. Burgermeister, and A. Eichberger. 2007. “Linearly Implicit Time Integration Methods in Real-Time Applications: DAEs and Stiff ODEs.” *Multibody System Dynamics* 17 (2-3): 99–117. https://doi.org/10.1007/s11044-007-9036-8

Blundell, M., and D. Harty. 2015. *The Multibody Systems Approach to Vehicle Dynamics*, 2nd edn. Amsterdam: Elsevier. https://doi.org/10.1016/C2012-0-07000-7

Burgermeister, B., M. Arnold, and A. Eichberger. 2011. “Smooth Velocity Approximation for Constrained Systems in Real-Time Simulation.” *Multibody System Dynamics* 26 (1): 1–14. https://doi.org/10.1007/s11044-011-9243-1

Bursi, O. S., A. Gonzalez-Buelga, L. Vulcan, S. A. Neild, D. J. Wagg, O. S. Bursi, A. Gonzalez-Buelga, L. Vulcan, S. A. Neild, and D. J. Wagg. 2008. "Novel Coupling Rosenbrock-Based Algorithms for Real-Time Dynamic Substructure Testing." *Earthquake Engineering & Structural Dynamics* 37 (3): 339–360. https://doi.org/10.1002/eqe.757

Butz, T., B. Simeon, and M. Stadler. 2010. "Optimal Design of Experiments for Estimating Parameters of a Vehicle Dynamics Simulation Model." *Journal of Computational and Nonlinear Dynamics* 5 (3): 031011-1–031011-7. https://doi.org/10.1115/1.4001391

Candreva, S., D. Mundo, M. Gubitosa, and A. Toso. 2014. "On the Correlation between a 3D High-Fidelity Multibody Vehicle Model and a 1D 15-DOFs Vehicle Concept Model." In ISMA International Conference on Noise and Vibration Engineering. Leuven, Belgium.

Carpinelli, M., M. Gubitosa, D. Mundo, and W. Desmet. 2016. "Automated Independent Coordinates' Switching for the Solution of Stiff DAEs with the Linearly Implicit Euler Method." *Multibody System Dynamics* 36 (1): 67–85. https://doi.org/10.1007/s11044-015-9455-x

Cibrario, V., M. Bäecker, J. De Cuyper, S. Donders, A. Toso, W. Prescott, A. Gallrein, and A. Gizatullin. 2014. "Real-Time Technologies in Driving Dynamics Performance Simulation through Full Vehicle Multibody, 1D Chassis Subsystems and Tire Modeling Integration." In *World Automotive Congress 2014*. Maastricht, The Netherlands.

Dahmen, W., and A. Reusken. 2008. *Numerik für Ingenieure und Naturwissenschaftler*, 2nd ed. Berlin, Heidelberg: Springer.

Ersoy, M., and S. Gies. 2017. *Fahrwerkhandbuch: Grundlagen - Fahrdynamik - Fahrverhalten - Komponenten - elektronische Systeme - Fahrerassistenz - autonomes Fahren - Perspektiven*, 5th ed. Wiesbaden: Springer Fachmedien.

Falchi, L., F. Goy, C. Ludwig, and A. Salgarello. 2022. "Feasibility Study for Ride Comfort Application by Functional Vehicle Models: New Excitation Method and Model Extension." In *12th International Munich Chassis Symposium 2021, ATZ Live*, edited byP. Pfeffer, 261–274. Berlin: Springer. https://doi.org/10.1007/978-3-662-64550-5_15

Hidalgo, A. F., and J. García de Jalón. 2015. "Real-Time Dynamic Simulations of Large Road Vehicles Using Dense, Sparse, and Parallelization Techniques." *Journal of Computational and Nonlinear Dynamics* 10 (3): 031005-1–031005-15. https://doi.org/10.1115/1.4028794

Hilber, H. M., T. J. Hughes, and R. L. Taylor. 1977. "Improved Numerical Dissipation for Time Integration Algorithms in Structural Dynamics." *Earthquake Engineering & Structural Dynamics* 5 (3): 283–292. https://doi.org/10.1002/eqe.4290050306

Intel. 2025. "Accelerate fast math with intel oneAPI math kernel library." https://www.intel.com/content/www/us/en/developer/tools/oneapi/onemkl.html#gs.91vex3

ISO 19365:2016. 2016. "Passenger cars - Validation of Vehicle Dynamic simulation - Sine with Dwell Stability Control Testing."

Jafarzadehpour, N., M. Bidlingmaier, and B. Corves. 2021. "Integrating Body-In-White Influences on Vehicle Dynamics into Real-Time Models." SAE Technical Paper Series 2021-01-5085. https://doi.org/10.4271/2021-01-5085

Kim, M., H. Song, and S.-S. Kim. 2019. "A Non-Iterative Implicit Integration Method Using a HHT-α Integrator for Real-Time Analysis of Multibody Systems." *Journal of Mechanical Science and Technology* 33 (3): 1087–1096. https://doi.org/10.1007/s12206-019-0208-2

Kim, S. S., and W. H. Jeong. 2009. "Real-Time Multibody Vehicle Model with Bush Compliance Effect Using Quasi-Static Analysis for HILS." *Multibody System Dynamics* 22 (4): 367–382. https://doi.org/10.1007/s11044-009-9160-8

Klomp, M., P. Sundström, and A. Johnsson. 2017. "Real-Time Simulation of Elasto-Kinematic Multi-Body Vehicle Models." In *Advanced Vehicle Control AVEC'16*, edited by J. Edelmann, M. Plöchl, P. E. Pfeffer, 255–260. Boca Raton: Crc Press. https://doi.org/10.1201/9781315265285-41

Kracht, F. E., and D. Schramm. 2020. "Real-Time Capable Calculation of Reaction Forces of Multibody Systems Using Optimized Bushings on the Example of a Vehicle Wheel Suspension." In *Multibody Dynamics 2019, Computational Methods in Applied Sciences Ser*, edited by A. Kecskeméthy, vol. 53, 409–416. Cham: Springer International Publishing AG. https://doi.org/10.1007/978-3-030-23132-3_49

Lee, J. K., J. S. Kang, and D. S. Bae. 2014. "An Efficient Real-Time Vehicle Simulation Method Using a Chassis-Based Kinematic Formulation." *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 228 (3): 272–284. https://doi.org/10.1177/0954407013507912

Lefèvre, S., F. E. Kracht, and D. Schramm. 2020. "Quasi-Static and Dynamic Suspension Measurements vs. Multi-Body and Real-Time Simulation Results." In *10th International Munich Chassis Symposium 2019, Proceedings*, 119–134. Wiesbaden, Germany: Springer Vieweg. https://doi.org/10.1007/978-3-658-26435-2_13

MathWorks. 2025. "Simulink coder documentation." https://de.mathworks.com/help/rtw/index.html?s_tid=CRUX_lftnav

MathWorks. 2025. "Solve Stiff Differential Equations—Trapezoidal Rule + Backward Differentiation Formula - MATLAB ode23tb." https://de.mathworks.com/help/matlab/ref/ode23tb.html

Prescott, W. 2014. "Real Time Simulation for High Fidelity Multibody Dynamics and Mechatronic Systems." In "Proceedings of the ASME International Mechanical Engineering Congress and Exposition - 2014." ASME, Montreal, Quebec, Canada. https://doi.org/10.1115/IMECE2014-39994

Rang, J. 2015. "Improved Traditional Rosenbrock–Wanner Methods for Stiff ODEs and DAEs." *Journal of Computational and Applied Mathematics* 286: 128–144. https://doi.org/10.1016/j.cam.2015.03.010

Rill, G. 2006. "A Modified Implicit Euler Algorithm for Solving Vehicle Dynamic Equations." *Multibody System Dynamics* 15 (1): 1–24. https://doi.org/10.1007/s11044-006-2359-z

Rill, G. 2019. "Multibody Systems and Simulation Techniques." In *Vehicle Dynamics of Modern Passenger Cars. CISM International Centre for Mechanical Sciences Ser*, edited by P. Lugner, vol. 582, 309–375. Cham: Springer. https://doi.org/10.1007/978-3-319-79008-4_6

Rosenbrock, H. H. 1963. "Some General Implicit Processes for the Numerical Solution of Differential Equations." *The Computer Journal* 5 (4): 329–330. https://doi.org/10.1093/comjnl/5.4.329

Rulka, W., and E. Pankiewicz. 2005. "MBS Approach to Generate Equations of Motions for HiL-Simulations in Vehicle Dynamics." *Multibody System Dynamics* 14 (3-4): 367–386. https://doi.org/10.1007/s11044-005-1144-8

Salaani, M. K., G. J. Heydinger, and P. A. Grygier. 2003. "Heavy Tractor-Trailer Vehicle Dynamics Modeling for the National Advanced Driving Simulator." *SAE Transactions* 112. https://doi.org/10.4271/2003-01-0965

Schmidt, S., E. Schmidt, W. Schick, and J. Henning. 2015. "Optimization of Vehicle Handling Performance Using a Full Vehicle Model with Multi-Body System (MBS) Suspensions in Multiple Real Time - Applying the DoE Method." *International Journal of Automotive Engineering* 6 (2): 53–57. https://doi.org/10.20485/jsaeijae.6.2_53

Schmitt, A. G. 2019. Real-time simulation of flexible multibody systems in vehicle dynamics. PhD Thesis, Technische Universität Hamburg, Institut für Mechanik und Meerestechnik. https://doi.org/10.15480/882.2510

Steihaug, T., and A. Wolfbrandt. 1979. "An Attempt to Avoid Exact Jacobian and Nonlinear Equations in the Numerical Solution of Stiff Differential Equations." *Mathematics of Computation* 33 (146): 521–534. https://doi.org/10.1090/S0025-5718-1979-0521273-8

U.S. Department of Transportation, National Highway Traffic Safety Administration. 2013. "Laboratory Test Procedure for Dynamic Rollover - The Fishhook Maneuver Test Procedure, New Car Assessment Program (NCAP)."

Zhang, H., Y. Duan, Y. Zhang, and J. Wu. 2025. "A Suspension Model Based on Semi-Recursive Method and Data-Driven Compliance Characteristics." SAE Technical Paper 2025-01-8282. https://doi.org/10.4271/2025-01-8282